Amendments to the Claims

Kindly amend claim 21 as follows:

1. (original) An apparatus within a pipelined microprocessor for forwarding store instruction results to a pipeline stage for execution of a load instruction, the apparatus comprising:

a result forwarding cache (RFC), for storing a plurality of store instruction results;

comparison logic, for comparing a load address of the load instruction with a plurality of store addresses associated with said plurality of store instruction results to generate an address match signal; and

control logic, configured to receive said match signal and selectively forward one of said plurality of store instruction results from said RFC to the pipeline stage in response to said match signal.

2. (original) The apparatus of claim 1, wherein said plurality of store instruction results comprise data to be stored from the microprocessor into a memory attached thereto.

3. (original) The apparatus of claim 1, wherein said load address specifies a location of data to be loaded into the microprocessor from a memory attached thereto.

4. (original) The apparatus of claim 1, wherein said RFC comprises a plurality of storage elements for storing a predetermined number of instruction results.

5. (original) The apparatus of claim 4, wherein said instruction results are received by said RFC from an execution unit of the microprocessor.

6. (original) The apparatus of claim 4, wherein said plurality of storage elements store said predetermined number of instruction results in a first-in-first-out manner.

7. (original) The apparatus of claim 4, wherein said predetermined number of instruction results is five.

8. (original) The apparatus of claim 1, wherein said load address and said plurality of store addresses comprise virtual addresses.

9. (original) The apparatus of claim 8, wherein said virtual addresses comprise x86 linear addresses.

10. (original) An apparatus for forwarding storehit data within stages of a pipelined microprocessor, the apparatus comprising:

a result forwarding cache (RFC), configured to forward a first plurality of store instruction results;

a data unit, configured to forward a second plurality of store instruction results; and

selection logic, coupled to said RFC and said data unit, for selectively providing one of said first and second plurality of store instruction results to a stage of the microprocessor pipeline executing a load instruction.

11. (original) The apparatus of claim 10, wherein said load instruction comprises a load address for specifying an address of data to be loaded into the microprocessor, wherein said selection logic is configured to forward one of said first and second plurality of store instruction results only if said load address matches one or more of a first and second plurality of store addresses corresponding to said first and second plurality of store instruction results.

12. (original) The apparatus of claim 11, wherein selection logic forwards said first plurality of store instruction results forwarded by said RFC at a higher priority than said second plurality of store instructions results forwarded by said data unit if said load address matches both one or more of said first plurality of store addresses and one or more of said second plurality of store addresses.

13. (original) The apparatus of claim 11, further comprising:

comparison logic, coupled to said selection logic, for comparing said load address with said first and second plurality of store addresses to determine whether

12. (original) The apparatus of claim 11, wherein selection logic forwards said first plurality of store instruction results forwarded by said RFC at a higher priority than said second plurality of store instructions results forwarded by said data unit if said load address matches both one or more of said first plurality of store addresses and one or more of said second plurality of store addresses.

13. (original) The apparatus of claim 11, further comprising:

comparison logic, coupled to said selection logic, for comparing said load address with said first and second plurality of store addresses to determine whether said load address matches one or more of said first and second plurality of store addresses.

14. (original) The apparatus of claim 11, wherein said data unit is configured to forward said second plurality of store instruction results from a plurality of store buffers of the microprocessor.

15. (original) The apparatus of claim 14, wherein said plurality of store buffers is configured to store said second plurality of store instruction results while said second plurality of store instruction results are written to a memory coupled to the microprocessor.

16. (original) The apparatus of claim 14, wherein said data unit is configured to forward a newest one of said second plurality of store instruction results if said load address matches more than one of said second plurality of store addresses.

17. (original) The apparatus of claim 11, wherein said RFC is configured to forward a newest one of said first plurality of store instruction results if said load address matches more than one of said first plurality of store addresses.

18. (previously presented) An apparatus for detecting storehit conditions in a pipelined microprocessor in a hierarchical manner, the apparatus comprising:

first comparison logic, for comparing a load instruction load address in a first stage of the pipeline with a first plurality of store addresses of first store instruction data in a plurality of stages of the pipeline subsequent to said first pipeline stage, wherein said plurality of stages of the pipeline subsequent to said first pipeline stage comprise non-store buffers;

second comparison logic, for comparing said load address with a second plurality of store addresses of second store instruction data in a plurality of store buffers of the microprocessor; and

control logic, coupled to said first and second comparison logic, configured to determine which of said first and second store instruction data is newest based on said first and second comparison logic comparing, and to forward said newest store instruction data to said first pipeline stage in response thereto.

19. (original) The apparatus of claim 18, wherein said first comparison logic is configured to compare virtual addresses.

20. (original) The apparatus of claim 18, wherein said second comparison logic is configured to compare physical addresses.

non-cacheable address regions of the microprocessor address space stored therein;

forwarding logic, for forwarding the storehit data from the first stage to the second stage during a first clock cycle; and

control logic, configured to receive said match signal and to assert a stall signal during a second clock cycle to stall the pipeline if the load address is within one of said plurality of non-cacheable address regions.

24. (original) The microprocessor of claim 23, further comprising:

a bus interface unit, for receiving data from a bus coupled to the microprocessor, said bus further coupled to a system memory and a plurality of peripheral devices; and

at least one response buffer, operatively coupled to the second stage, for receiving load data specified by the load address from said bus interface unit, and for providing said load data to the second stage to replace the storehit data if the load address is within one of said plurality of non-cacheable address regions.

25. (original) The microprocessor of claim 23, wherein said plurality of non-cacheable regions stored in said address region logic are software-programmable.

26. (original) A method for forwarding storehit data in a microprocessor pipeline, the method comprising:

detecting a storehit condition, wherein a load instruction in a stage of the pipeline specifies data generated by a previous store instruction, wherein said data is still present in the pipeline;

determining whether said data is present in a result forwarding cache of the microprocessor;

selectively forwarding said data from said result forwarding cache to said stage if said data is in said result forwarding cache; and

selectively forwarding said data from a data unit of the microprocessor to said stage if said data is not in said result forwarding cache.

27. (original) The method of claim 26, further comprising:

storing results data of each store instruction executed by an execution unit of the microprocessor in said result forwarding cache.

28. (original) The method of claim 26, wherein said detecting said storehit condition comprises:

comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in the pipeline below said stage; and

determining said address matches one or more of said plurality of data addresses.

29. (original) The method of claim 26, wherein said determining whether said data is present in said result forwarding cache comprises:

comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in a predetermined number of stages of the pipeline below said stage;

wherein said predetermined number equals a number of result entries in said result forwarding cache.

30. (original) A method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:

speculatively forwarding storehit data from a first stage to a second stage of the pipeline based on a virtual address comparison between a load address and a plurality of store addresses;

detecting a virtual aliasing condition with respect to said load address and one of said plurality of store addresses based on a physical address comparison between said load address and said plurality of store addresses after said speculatively forwarding; and

stalling the pipeline in response to said detecting said virtual aliasing condition.

31. (original) The method of claim 30, further comprising:

forwarding correction data from a third stage of the pipeline to said second stage after said stalling the pipeline; and

25. (original) The microprocessor of claim 23, wherein said plurality of non-cacheable regions stored in said address region logic are software-programmable.

26. (previously presented) A method for forwarding storehit data in a microprocessor pipeline, the method comprising:

storing at least one store instruction result and at least one non-store instruction result into a result forwarding cache of the microprocessor;

detecting a storehit condition, wherein a load instruction in a stage of the pipeline specifies data generated by a previous store instruction, wherein said data is still present in the pipeline;

determining whether said data is present in said result forwarding cache;

selectively forwarding said data from said result forwarding cache to said stage if said data is in said result forwarding cache; and

selectively forwarding said data from a data unit of the microprocessor to said stage if said data is not in said result forwarding cache.

27. (original) The method of claim 26, further comprising:

storing results data of each store instruction executed by an execution unit of the microprocessor in said result forwarding cache.

28. (original) The method of claim 26, wherein said detecting said storehit condition comprises:

comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in the pipeline below said stage; and

determining said address matches one or more of said plurality of data addresses.

29. (original) The method of claim 26, wherein said determining whether said data is present in said result forwarding cache comprises:

comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in a predetermined number of stages of the pipeline below said stage;